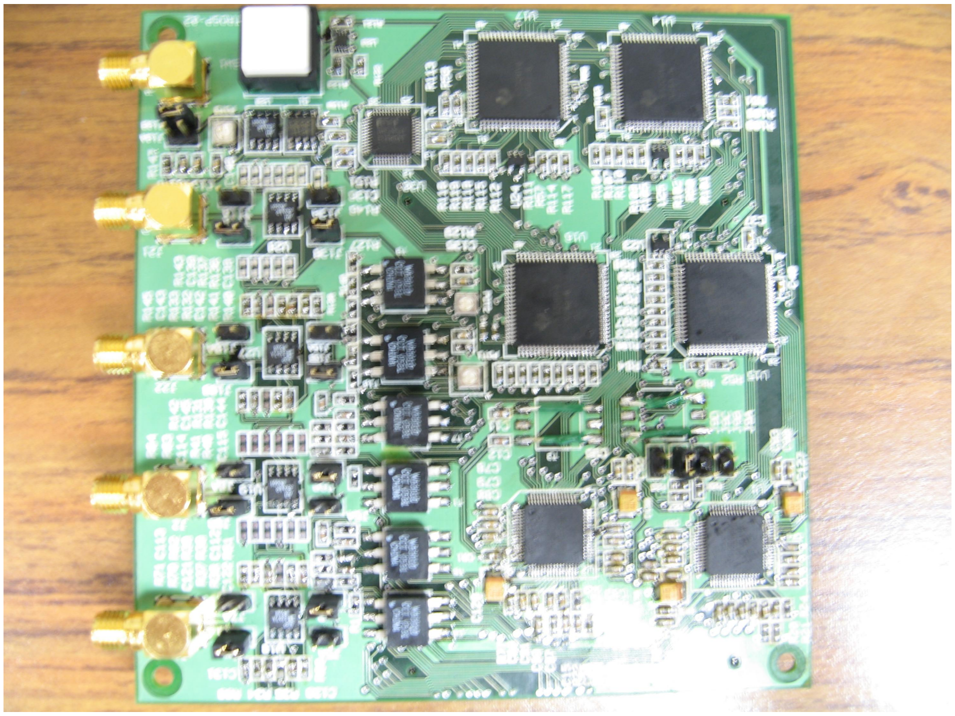
Signal Analysis Daughter Card

USER'S MANUAL



Electrical & Computer Engineering
North Dakota State University
1411 Centennial Boulevard
Fargo, ND 58105
Phone: (701) 231-7019

Table of Contents

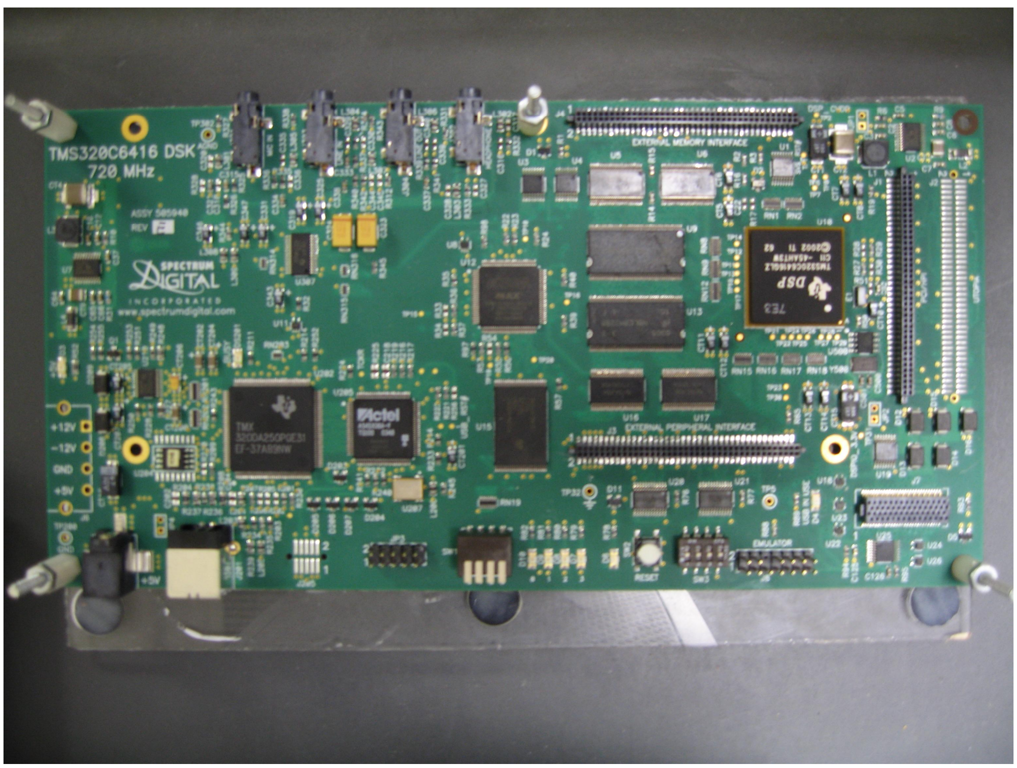
1 General Overview

1.1 Introduction
This design is the implementation of a signal analysis concept that was developed at North Dakota State University. The Signal Analysis Daughter Card is capable of interfacing to a Spectrum Digital 6416 Digital Signal Processor Starter Kit (DSK). This DSK utilizes the power of the TI TMS320C6416 Digital Signal Processor (DSP) in order to acquire, process, and transmit analog signals at very high frequencies. This document explains the general use of the Signal Analysis Daughter Card.

1.2 User Requirements
In order to use this Signal Analysis Daughter Card you will need the following:

A Spectrum Digital 6416 DSP Starter Kit
Code Composer Studio Software

Figure 1.1 Spectrum Digital DSK



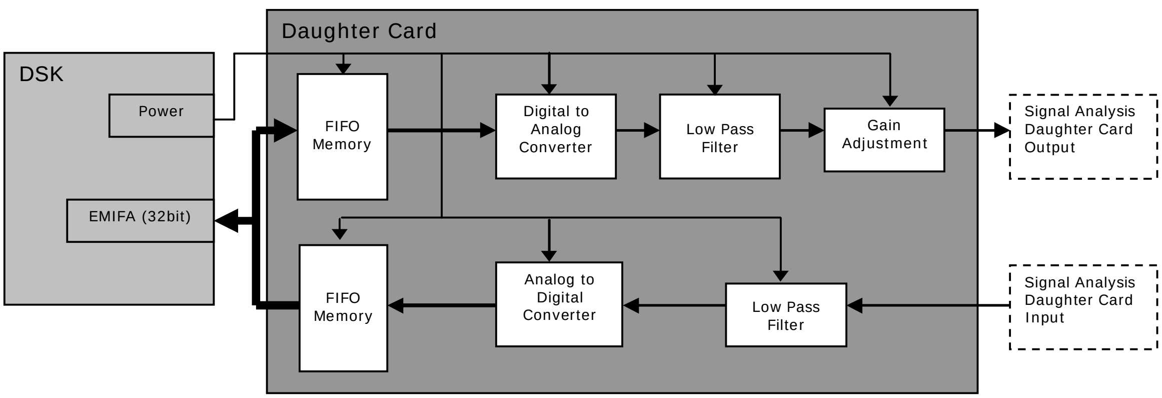
2 Hardware Overview

2.1 General Hardware Overview

Each of the two channels of the Signal Analysis Daughter Card is designed to do the following (in order):

1. Acquire an analog signal via the SMA connector.
2. Filter out the high frequency content.
3. Convert the analog signal to a digital signal.
4. Buffer the digital signal in a temporary memory storage device.
5. Send the digital signal from memory into the DSK for manipulation.
6. Send the manipulated digital signal from the DSK into another temporary memory storage device.
7. Convert the digital signal back to an analog signal.
8. Filter and apply a gain to the analog signal.
9. Transmit the manipulated signal out via SMA connector.

Figure 2.1 Block Diagram of Signal Analysis Daughter Card



2.2 Optional Hardware Configurations

All configurable settings on the daughter card are connected via pin connectors and jumpers. The three changes that can be made are:

1. ADC output data
2. Filter bypass
3. DAC output reference

1) The ADC output mode can be changed to either straight binary or 2's compliment and from either data rising on falling or rising edge. This change can be made by jumping the two pins as detailed in the table below.

Figure 2.2 ADC Output Mode Configuration

Jumper	Data Format/Clock Output Polarity
J9D	Straight Binary/Data valid on rising edge
J9C	Two's Complement/Data valid on rising edge
J9B	Straight Binary/Data valid on falling edge
J9A	Two's Complement/Data valid on falling edge

2) The filters can be bypassed by changing the jumpers around them.

Note: The channels can either be bypassed or filtered, not both.

Figure 2.3 Optional Filter Bypass

Channel	Jumpers
Input 1 – Filtered	J7A, J5A
Input 1 – Bypassed	J7B, J5B
Input 2 – Filtered	J8A, J6A
Input 2 – Bypassed	J8B, J6B
Output 1 – Filtered	J10A,J15A
Output 1 – Bypassed	J10B,J15B
Output 2 – Filtered	J11A,J13A
Output 2 – Bypassed	J11B, J13B

3) The Digital to Analog Converter also contains a reference voltage that it bases its conversion to analog on. At default, it is connected to a grounded capacitor. It can also be connected to a circuit whose voltage is determined by a variable resistor. Finally, there is a circuit that can be connected to input an external voltage via SMA connector.

Figure 2.4 Optional DAC Voltage References

J19A left open	DAC uses default internal reference
J19A jumped together	Reference controlled by potentiometer
J19A jumped to J19B	Reference controlled by external source via SMA

3 Hardware Installation

3.1 Installing the Signal Analysis Daughter Card

Follow these easy instructions to install the Signal Analysis Daughter Card:

1. With the power cord not connected to the DSK, gently push the daughter card onto the 6416 DSK. The daughter card should fit so that the SMA connectors face to the edge of the board. Make sure to support the DSK in order to minimize the flexing of the DSK.
2. Once the daughter card is attached to the DSK, plug the power cord into the wall and connect the USB cable to the computer that contains Code Composer Studio. Do not yet start Code Composer Studio.
3. Plug the USB cable into the appropriate connector on the DSK. Once this is done, plug the power cord into the appropriate slot.

WARNING FROM THIS POINT ON, THE DSK AND THE DAUGHTER CARD ARE BEING FED POWER. USE CAUTION AND AVOID TOUCHING ANYTHING ON EITHER BOARD THAT MAY CAUSE AN ELECTRICAL SHOCK OR SHORT THE BOARD OUT.

4. Once all connections are made to the board, open Code Composer Studio. The details of this part are found in the next section.

4 Software Configuration

4.1 Loading Programs

Follow these easy steps to begin running the Signal Analysis Daughter Card:

1. Make sure that the hardware has been properly installed.
2. Open Code Composer Studio (CCS) by double clicking on the desktop icon.
3. Once CCS is running, open the desired program (*.out) by clicking File → Load Program
4. After you see the program load, click the "Run" button, press F5, or go to Debug → Run to run the program. To stop the program press Shift-F5, or go to Debug →Halt.
5. Enjoy the Signal Analysis Daughter Card.

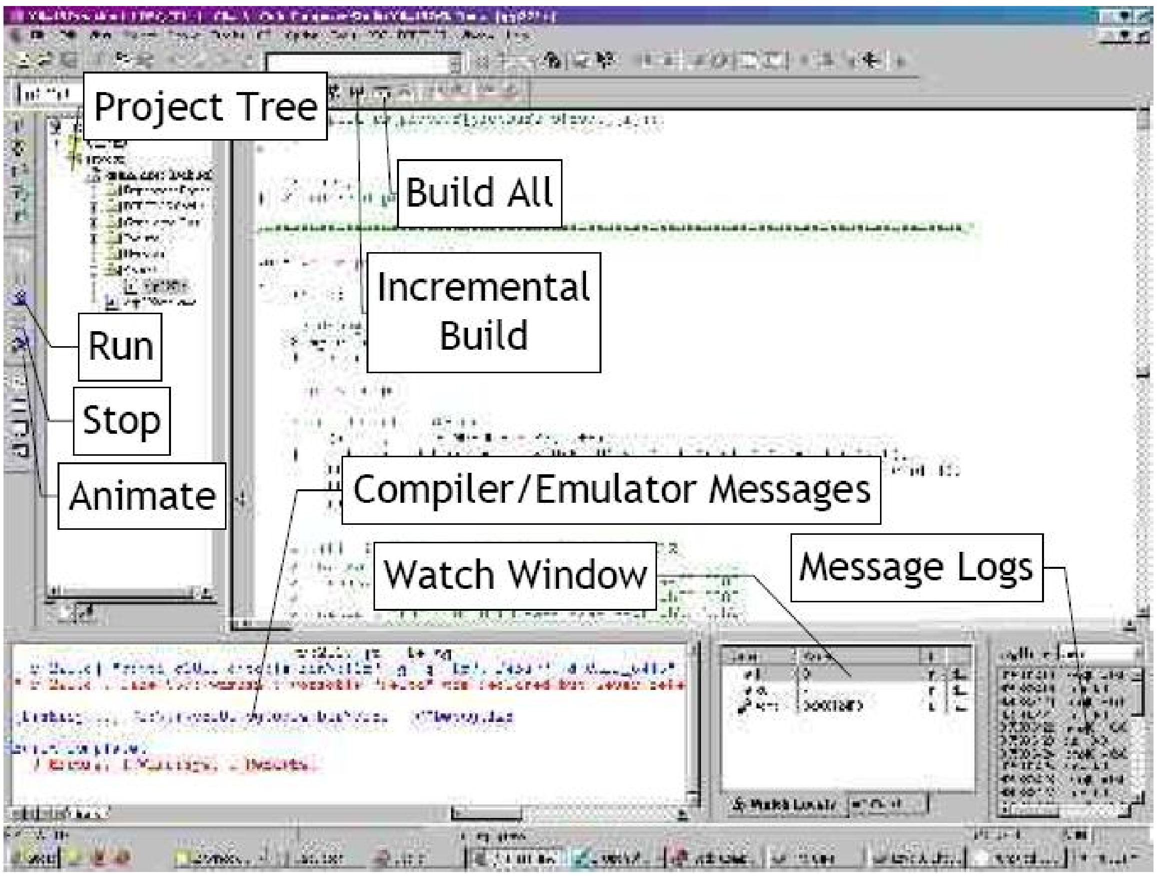
4.2 Editing Software**

Follow these easy steps to begin editing the software:

1. Make sure that the hardware has been properly installed.
2. Open Code Composer Studio (CCS) by double clicking on the desktop icon.
3. Once CCS is running, open the desired project (*.pjt) by clicking Menu → Project → Open. The project should be displayed on the left hand window.
4. Click the "+" next to the project name to expand the project.

5. Scroll down the project tree to the "Sources" node and expand that the same way. It is here that you will find the main source code for the desired project.
6. When done editing** the code, rebuild (compile & link) the project. This can be done by clicking Project → Build All. This will create the proper "*.out" file to load. An Incremental Build can also be done. This will just rebuild the edited portion of the source code. To perform this, just click the Incremental Build button on the toolbar.
7. Follow the above instructions on how to load the edited program.

Figure 4.1 Code Composer Studio Screen Shot



**** Edit at your own risk**

4.3 Created Programs

Three programs have been created for use with the Signal Analysis Daughter Card:

1. Read_Test.out
   The read test simply reads the input into the DSK.

2. Write_Test.out
   The write test creates ramp function in the DSP and sends it out to the daughter card.

Figure 4.2 Write Test Created Ramp function

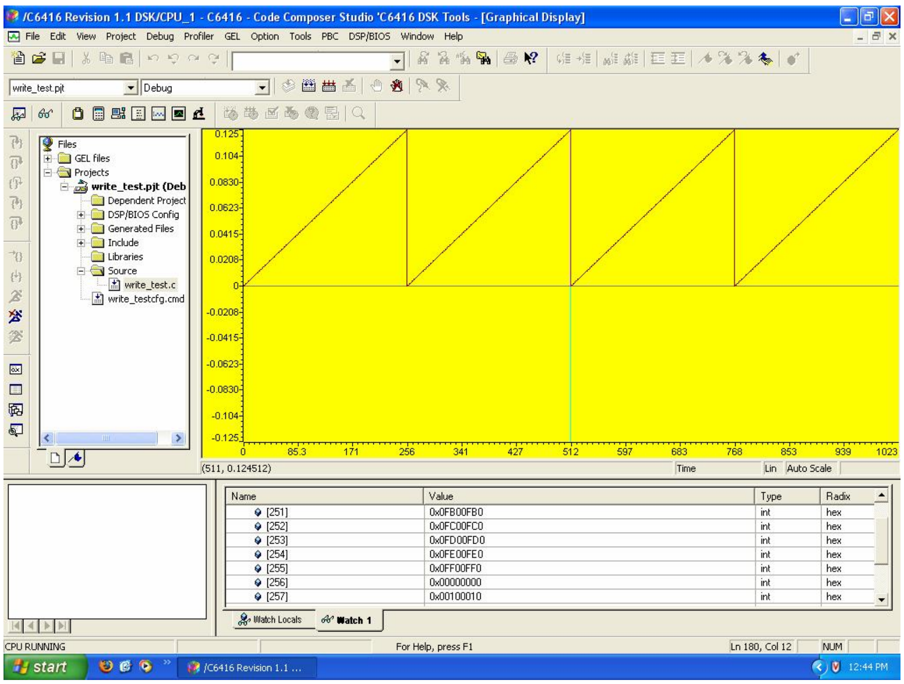
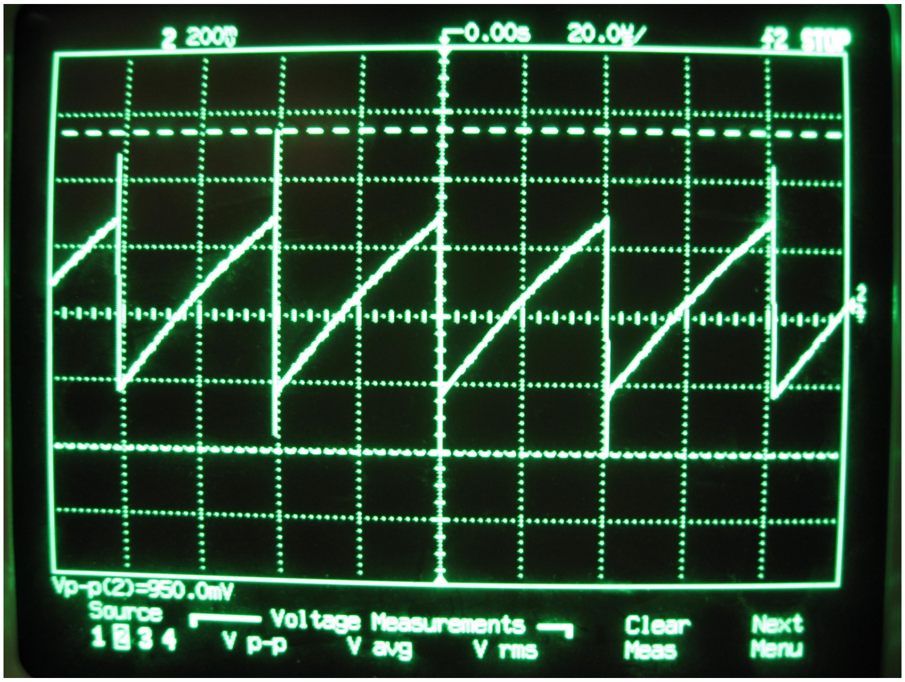


Figure 4.3 Write Test Ramp Function Output



3. Timereversal.out
   This program is an attempt to use the Signal Analysis Daughter Card as a time reversal platform.

5 Analyzing the Daughter Card

5.1 Overview

Once the Signal Analysis Daughter Card has been installed and the proper program has been loaded, it is appropriate to look at some signals in order to check signal integrity. To do this, use an oscilloscope and probe the points of interest.

5.2 Important Lines to Observe

- TOUT[0] & [1]
  - These two lines determine if the program has loaded correctly.
  - They are the triggers to the ADC and the DAC.

- ADC_CLK_OUT[0] & [1]
  - These lines are used to synchronize the output clock to the output data.
  - They tell whether the ADC is running or not.

- AARE# & AAWE#
    - These clocks are the asynchronous clock for the EMIF.

- DATA OUT LINES
    - These lines are either on the pins of the ADC & DAC, or interconnecting vias throughout the board.
    - It is appropriate to check these lines in order to see if there is valid data on the bus lines.

- POWER AND GROUND
    - It is important to check that each component is powered and grounded appropriately.

Figure 5.1 Clocks

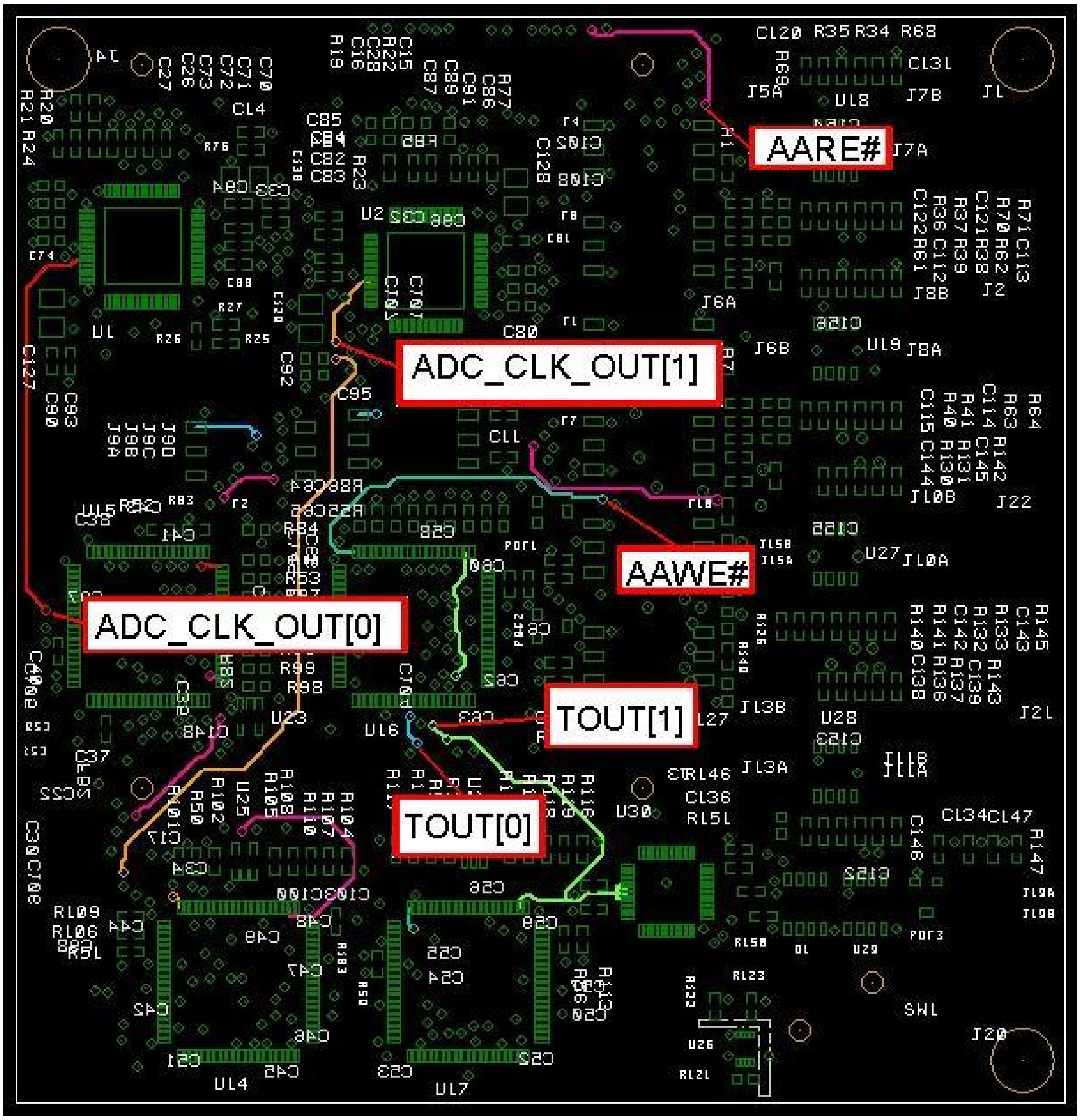


Figure 5.2 Analog to Digital Data & Digital to Analog Data

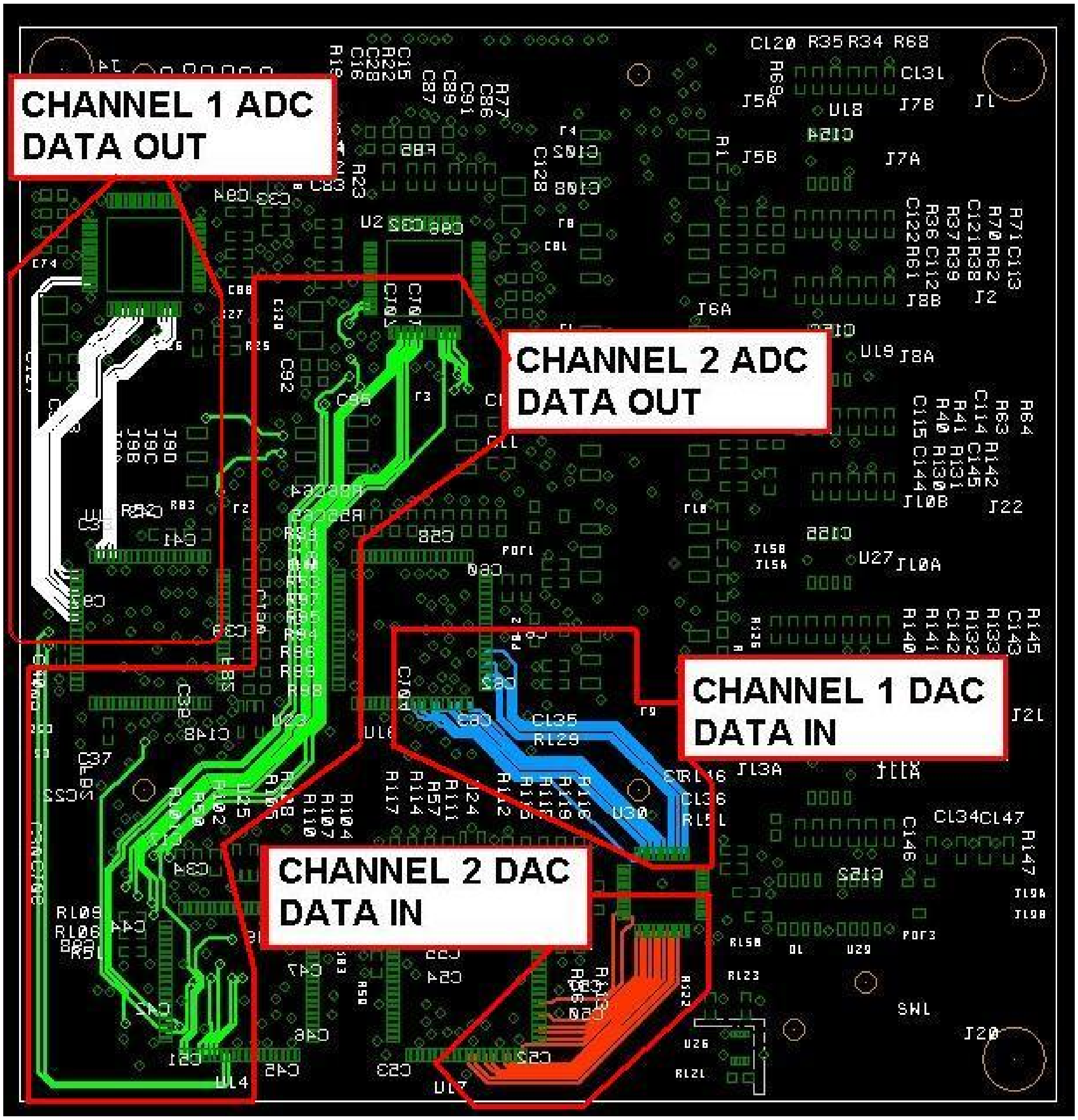
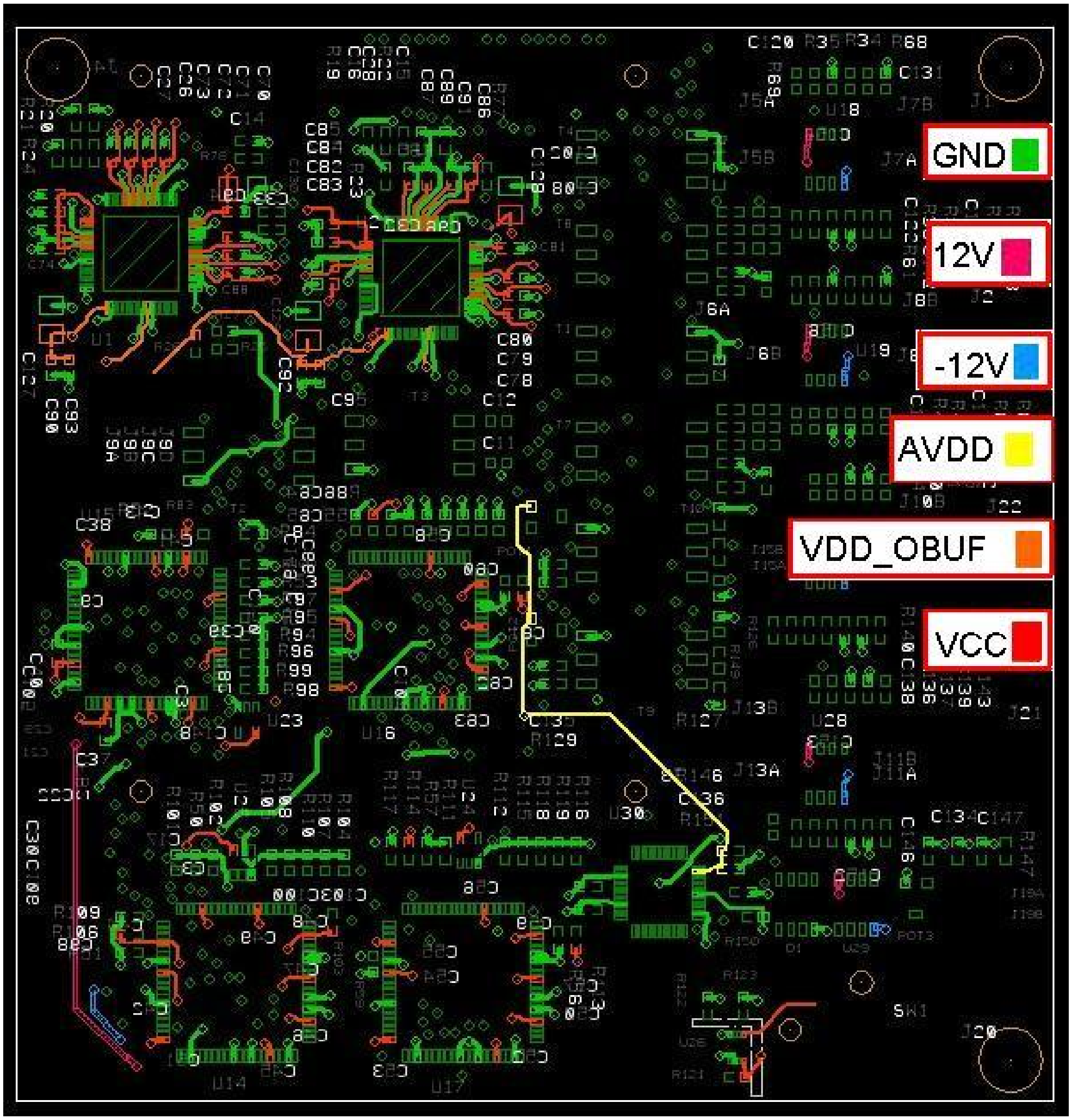


Figure 5.3 Power Supplies and Ground



6 Additional Information

6.1 Contact & Assistance

For any assistance with the Signal Analysis Daughter Card that is not mentioned in this user's manual, please see the Technical Information Paper on this project. Here you will find a more in depth explanation of the hardware and software. You will also find references to the schematic, the PCB, the components, and design analysis.

For any other assistance not in these documents, please contact the following:

Jonathan Blixt
<Jon.Blixt@ndsu.edu>

David Hinkemeyer
<David.Hinkemeyer@ndsu.edu>

Kurt Pulczinski
<Kurt.Pulczinski@ndsu.edu>

Dr. David Farden
Department of Electrical & Computer Engineering
North Dakota State University
1411 Centennial Boulevard
Fargo, ND 58105
Phone: (701) 231-7019